In the *Proceedings of the Seventh International Conference on Machine Learning*, Austin, TX, 1990.

Explanations of Empirically Derived Reactive Plans

Diana F. Gordon (gordon@aic.nrl.navy.mil) **John J. Grefenstette (gref@aic.nrl.navy.mil)**

Navy Center for Applied Research in Artificial Intelligence
Naval Research Laboratory, Code 5514
Washington, D.C. 20375-5000

Abstract

Given an adequate simulation model of the task environment and payoff function that measures the quality of partially successful plans, competition-based heuristics such as genetic algorithms can develop high performance reactive rules for interesting sequential decision tasks. We have previously described an implemented system, called SAMUEL, for learning reactive plans and have shown that the system can successfully learn rules for a laboratory scale tactical problem. In this paper, we describe a method for deriving explanations to justify the success of such empirically derived rule sets. The method consists of inferring plausible subgoals and then explaining how the reactive rules trigger a sequence of actions (i.e., a strategy) to satisfy the subgoals.

1 Introduction

This report is part of an on-going study concerning learning reactive plans for sequential decision tasks given a simulation of the task environment. In particular, we have been investigating techniques that allow a learning system to actively explore alternative behaviors in simulation, and to construct high performance rules from this experience using competition-based methods. Our current research focuses on learning reactive rules for a variety of tactical scenarios. Learning tactical rules is especially difficult if the environment is only partially modeled, contains other independent agents, or permits only limited sensing of important state variables. Such features reduce the utility of traditional projective problem solving (Mitchell, 1983; Minton et. al, 1989) and favor the use of reactive control rules that respond to current information and suggest useful actions (Agre and Chapman, 1987; Schoppers, 1987). We have been investigating the usefulness of genetic algorithms and other competition-based heuristics (Grefenstette, 1988) to learn high performance reactive rules in the absence of a strong domain theory. The approach has been implemented in a system called SAMUEL (Grefenstette, 1989). One of the important differences between SAMUEL and many other genetic learning systems is that SAMUEL learns rules expressed in a high level rule language. The use of a symbolic rule language is intended to facilitate the incorporation of more powerful learning methods into the system where appropriate. In this paper, we investigate the use of explanation-based learning methods to explain the success of the empirically learned plans found by the genetic learning system, and to suggest possible improvements.

SAMUEL consists of three major components: a problem specific module, a performance module, and a learning module. The problem specific module consists of the task environment simulation, or world model, and its interfaces. The performance module consists of a competition-based production system that performs matching, conflict resolution and credit assignment. The learning module uses a genetic algorithm to develop high performance reactive plans, each plan expressed as a set of condition-action rules. Each plan is evaluated by testing its performance in controlling the world model through the performance module. Genetic operators, such as crossover and mutation, produce plausible new plans from high performance precursors.

Experiments have shown that SAMUEL learns highly effective reactive plans for laboratory scale tactical problems (Grefenstette, 1989). However, even though the individual rules of a plan can be interpreted, the strategy underlying the plan is often not apparent. We are currently expanding our focus

to include the derivation of explanations of SAMUEL's reactive rules. These explanations are expected to clarify the system's performance to system users as well as to generate new reactive rules for SAMUEL.

In this paper, we first discuss a simulated environment to which SAMUEL has been successfully applied. The remainder of the paper is devoted to describing our research on the topic of generating explanations of reactive plans.

This work is part of an on-going study of genetic algorithms for learning tactical plans. The current system is detailed in (Grefenstette, Ramsey & Schultz, 1990). An analysis of the credit assignment methods in appears in (Grefenstette, 1988). A study of the effects of sensor noise on appears in (Schultz, Ramsey & Grefenstette, 1990).

2 The Evasive Maneuvers Problem

We have tested SAMUEL initially in the context of a particular task called Evasive Maneuvers (EM), inspired in part by (Erickson and Zytkow, 1988). In the EM simulation, there are two objects of interest, a plane and a missile, which maneuver in a two-dimensional world. The object is to control the turning rate of the plane to avoid being hit by the approaching missile. The missile tracks the motion of the plane and steers toward the plane's anticipated position. The initial speed of the missile is greater than that of the plane, but the missile loses speed as it maneuvers. If the missile speed drops below some threshold, it loses maneuverability and drops out of the sky. It is assumed that the plane is more maneuverable than the missile, that is, the plane has a smaller turning radius.

There exist six sensors that provide information about the current tactical state:

1) *last-turn*: the current turning rate of the plane. This sensor can assume nine values, ranging from -180 degrees to 180 degrees in 45 degree increments.
2) *time*: a clock that indicates time since detection of the missile. Assumes integer values between 0 and 19.
3) *range*: the missile's current distance from the plane. Assumes values from 0 to 1500 in increments of 100.
4) *bearing*: the direction from the plane to the missile. Assumes integer values from 1 to 12. The bearing is expressed in "clock terminology", in which *12 o'clock* denotes dead ahead of the plane, and *6 o'clock* denotes directly behind the plane.
5) *heading*: the missile's direction relative to the plane. Assumes values from 0 to 350 in increments of 10 degrees. A heading of 0 indicates that the missile is aimed directly at the plane's current position, whereas a heading of 180 means the missile is aimed directly away from the plane.
6) *speed*: the missile's current speed measured relative to the ground. Assumes values from 0 to 1000 in increments of 50.

In addition to the sensors, there is one control variable, namely, the plane's *turning-rate*. Turning-rate has nine possible values, between -180 and 180 degrees in 45 degree increments. The learning objective is to develop a set of decision rules that map current sensor readings into actions that successfully evade the missile whenever possible. The rule condition contains sensor ranges (which may be cyclic), and the action specifies a setting for the control variable. An example of an actual decision rule learning by SAMUEL is the following:

```
RULE 16:
IF      (and (last-turn [-135, 135]) (time [2, 12])
        (range [0, 700]) (bearing [2, 6])
        (heading [0, 30]) (speed [100, 950]))
THEN    (turn 90)
STRENGTH      949
```

The EM process is divided into episodes that begin with the missile approaching the plane from a randomly chosen direction and that end when either the plane is hit or the missile velocity falls below a given threshold. The critic module provides numeric feedback at the end of each episode that measures the extent to which the missile has been successfully evaded. In the case of unsuccessful evasion, partial credit is given reflecting the plane's survival time (see (Grefenstette et. al, 1990)). Each decision rule is assigned a numeric *strength* that serves as a prediction of the rule's utility. The system uses incremental credit assignment methods (Grefenstette, 1988) to update the rule strengths based on feedback from the critic received at the end of the episode. Experiments have shown that SAMUEL can learn high-performance rule sets (plans) for this task (Grefenstette, 1989).

As can be seen from the above example, while the rules are individually understandable, the underlying strategy behind the rules is not usually clear from inspection. On the other hand, a person

who watches a display of the EM task under the control of the learned rules can usually describe the strategy being followed in conceptual terms, for example:

> *Get the missile directly behind the plane, let it get fairly close, then make a hard left turn.*

Once such a description has been obtained, qualitative reasoning can be applied to explain and justify the strategy. It is expected that explanation-based methods will help to explicate the higher-level strategies being learned, making the results of the empirically learning more easily accepted by human operators and, ultimately, expediting the learning process itself. The remainder of the paper offers initial steps in this direction.

3 Explaining Empirically Derived Rules

Our approach to applying explanation-based techniques to reactive plans can be divided into four phases:

(1) inferring plausible subgoals;
(2) confirming subgoal satisfaction;
(3) creating explanations for reactive plans; and
(4) deriving new rules.

The following sections elaborate our approach to each of the first three phases. The fourth phase is outlined under our plans for future research.

3.1 Inferring Plausible Subgoals

Prior to deriving explanations that SAMUEL's actions are intended to satisfy particular subgoals, the system first attempts to derive plausible subgoals, such as "increase range to missile" or "increase missile deceleration" from a trace of the behavior of the system under the control of the learned rules. A trace covering the actions occurring over a single episode is examined. Traces consist of snapshots of sensor readings followed by the decision rule that has fired. Each snapshot is associated with a time, or state. An example of a trace is shown in Figure 1, where "lturn", "brng", and "hdng" are abbreviations for last turn, bearing, and heading. The action is the turn taken by the plane at this time. In order to simplify the trace shown here, the decision rules do not appear.

A domain theory has been developed for automating subgoal derivation. This part of the domain theory consists of *plausible subgoal derivation* (PSD) rules such as the following:

lturn	time	range	brng	hdng	speed	action
0	0	1000	7	0	700	0
0	1	600	7	0	650	135
135	2	0	9	350	550	0
0	3	300	3	290	400	45
45	4	200	6	0	300	-135
-135	5	100	4	20	250	90
90	6	100	7	0	200	0
0	7	300	6	0	200	45
45	8	400	7	0	150	45
45	9	500	8	0	150	45
45	10	500	8	0	100	-90
-90	11	600	5	0	100	-45
-45	12	700	4	0	100	45

Fig. 1. Example execution trace.

PSD 1: IF range(m) > RANGE1
THEN PLAUSIBLE-SUBGOAL
(INCREASING deceleration(m))

PSD 2: IF range(m) < RANGE2
THEN PLAUSIBLE-SUBGOAL
(INCREASING range(m))

where RANGE1 and RANGE2 are user-definable parameters and m represents the missile. The trace is examined to find the first time at which a PSD rule precondition, such as "range(m) > RANGE1", holds.

The algorithm for finding plausible subgoals is the following:

PSD ALGORITHM: Find the set of all time intervals in the execution trace of an episode for which the sensor values satisfy the PSD rule condition during that interval. This set, called the *trigger set*, consists of situations that would plausibly trigger the implementation of a strategy to satisfy the subgoal specified in the PSD rule.

In the example trace above, if RANGE1 were set to 900, then there is one time interval (of length one unit) that satisfies the condition for PSD1. This interval is [0,0] and, therefore, the trigger set is simply { [0,0] }. Since PSD1 is satisfied, its subgoal,

namely, ''(INCREASING deceleration(m))'', is proposed as a candidate subgoal.

Once a plausible subgoal is found, the next task is to determine whether the subgoal has been satisfied. Satisfaction is determined by applying the confirmation procedure described in the next section for time intervals in the trigger set until either the set of intervals is exhausted or the subgoal has been confirmed.

3.2 Confirming Subgoal Satisfaction

Subgoal satisfaction is determined by once again scanning the execution trace. Scanning begins at the time in the trace following a time interval from the trigger set. Subgoal confirmation requires an additional domain theory. In this case, SAMUEL's decision rule language is extended to capture further information from the trace. For example, the system extracts from the trace information about the *change* in sensor values over time. The speed or range of the missile, for instance, may increase from one state to the next. By scanning the trace over multiple states, the system derives acceleration and range increase information for confirming subgoal satisfaction.

The confirmation of subgoal satisfaction begins when a time interval is chosen from the trigger set. In the current implementation, the user defines a window over which the subgoal satisfaction check is executed. The window begins at a user-defined time that is after the trigger set time interval. Continuing with the example above, suppose the system must confirm that the increasing missile deceleration goal has been achieved over the time window that extends from time 1 to time 3. Then the change in missile speed over this interval is checked to be certain that missile deceleration is increasing. The deceleration is increasing from 100 to 150 over this time interval. Therefore, subgoal satisfaction has been confirmed.

Once subgoals have been derived and confirmed, explanations may be generated to justify the observed behavior. The next section describes the process of explanation generation.

3.3 Creating Explanations

After deriving plausible subgoals and confirming that they are satisfied, explanations may be formed which prove that sequences of SAMUEL's decision rules satisfy the subgoals. Explaining failure to satisfy subgoals is presented as future work.

Creating justifications for successful subgoal satisfaction requires the development of a domain theory that captures important results of particular actions. We are adapting Forbus's Qualitative Process Theory (Forbus, 1984) for the interpretation of the empirically derived rules similarly to the way this theory is adapted in (Gervasio, 1989). Qualitative Process Theory (QP Theory) expresses common sense notions about qualitative relationships between objects.

We are currently using QP Theory to define *processes* relevant to EM. A process is defined in (Forbus, 1984) as something that acts through time to change the parameters of objects in a situation. Example processes are fluid and heat flow, boiling, and motion. We define an EM process below. The individuals are the objects on which the process acts. The quantity conditions are inequalities regarding the quantities of individuals that can be predicted solely within dynamics. Preconditions are conditions that must hold during the process but which need not be predictable using dynamics. Relations are statements that are true during the process. A process is *active* whenever its preconditions and quantity conditions hold. The Q+/Q- relations define qualitative proportionalities. (Q+ X, Y) means that parameter X is directly proportional to parameter Y. (Q- X, Y) means that X and Y are inversely proportional.

process missile-evasion (p, m)

Individuals:
p, a plane
m, a missile

Quantity Conditions:
speed(p) > 0
speed(m) > 0

Preconditions:
range(m) > 0

Relations:
(Q+ deceleration(m), turning-rate(m))
(Q+ turning-rate(m), turning-rate(p))
(Q- speed(p), turning-rate(p))
(Q- turning-rate(m), range(m))

The above process description is incomplete and is not entirely accurate. Since we do not intend

to engineer a complete and perfect domain theory, our system will eventually possess a capability to diagnose errors in its domain theory.

Once a partial domain theory exists, it is possible to create plausible explanations of the events that occurred during an EM episode. Explanations are derived by creating proofs using the process relations similarly to (Gervasio, 1989). The proof begins with an observable but noncontrollable subgoal and terminates when a change in a controllable parameter has been found that is believed to have caused subgoal satisfaction. The body of the proof consists of QP Theory relational rules, such as those presented above. For example, the following proof explains how the increasing turning-rate of the plane eventually causes the missile deceleration to increase.

(EXPLANATION
 (INCREASING deceleration(m))
 ((Q+ deceleration(m) turning-rate(m))
 (Q+ turning-rate(m) turning-rate(p))
 (INCREASING turning-rate(p))))

The above proof has terminated with a statement that the plane turning rate is increasing. (The plane turning rate is currently the only controllable parameter.) The increasing turning rate is hypothesized as having initiated a strategy to achieve subgoal satisfaction. The system next verifies (by examining the execution trace) that this behavior has, in fact, occurred. For the above example, this would consist of a check to be certain that the plane turning rate is increasing during the time period that begins during the trigger set time interval and ends at some user-specified time following this interval. In the example trace above, the condition that the turning rate must be increasing would be satisfied if the plane's actions were examined from time 0 to time 1.

The selection of times for checking both subgoal satisfaction and triggering behaviors is currently done by the user. These are important parameters, yet they are difficult to choose. We next describe our plans for future work. These plans include automating the choice of these parameters, as well as other parts of the system.

4 Future Work

There are a few important directions that we plan to pursue. The first direction consists of ordering explanations according to their degree of plausibility. The second direction consists of using the explanations to generate new decision rules for SAMUEL. Third, we plan to automate the generation of system parameters and rules. The fourth future direction consists of diagnosing failures. Finally, we would like to increase the complexity of the EM problem.

Currently, we are running experiments to determine the differences in the degree of plausibility of various explanations. The manner in which this is being done is by generating explanations from multiple episode traces. From our experiences with explanation generation, we have been observing that some explanations/subgoals are considered plausible more frequently than others. We plan to use this information about the frequency to order the PSD rules in a manner that reflects the plausibility of explanations, e.g., more plausible subgoals are tried first.

The second direction for future research consists of generating new decision rules from the explanations. If a subgoal is satisfied, and an explanation is generated for subgoal satisfaction, then the system can generalize the explanation (perhaps using the explanation-based learning methods of (Mitchell, Keller & Kedar-Cabelli, 1986)) and then use the generalized explanation to generate new decision rules. Given a successful explanation, SAMUEL's performance can benefit by the creation of new decision rules that are expected to achieve the same results as the rules from which the explanation is formed. The process of generating decision rules from generalized explanations is one of rule specialization. We are currently considering using ideas from MARVIN (Sammut and Banerji, 1986) for designing the rule specialization process. Once new decision rules have been created, they can be fed back into SAMUEL's performance module to augment the existing rule sets. These modified rule sets may then be empirically evaluated using the EM simulator.

The third direction planned for our research is the automation of certain portions of the system that are currently provided by the user. For example, system parameters, such as the user-input window size for subgoal confirmation, might be empirically determined. Furthermore, the domain theory might also be derived empirically. For instance, the Q+/- relationships in the domain theory for explanations could be extracted from the execution traces.

Although we have been able to generate explanations for successful subgoal satisfaction, a ripe area for future research is the addition of the

ability to handle failures. If the system derives an explanation that the reactive rules are intended to achieve a particular subgoal, but the trace does not verify that the subgoal has been satisfied, then there exist four possible cases:

(1) The chosen explanation is incorrect, but the domain theory is not faulty
(2) The plausible subgoal that is inferred is not actually the subgoal that the system is trying to achieve
(3) The reactive rules are intended to achieve a subgoal, but the system has encountered some unexpected interference
(4) The domain theory is incorrect or incomplete

Although the generation of alternative explanations would be a relatively simple a solution for the first case, the other cases would require more sophisticated error diagnosis.

A final direction for future research is to increase the complexity of the EM problem. For example, the only controllable parameter currently implemented is the plane turning rate. More controllable parameters might be added. Furthermore, the problem difficulty would be greatly increased if the number of missiles were increased. Ultimately, we would like SAMUEL to be able to handle realistic problems.

5 Summary

Progress in generating and using explanations of reactive plans for SAMUEL is expected to provide an important step toward reducing the burden placed on the system's empirical learning mechanisms. The eventual goal of our research is to use these explanations to create high performance reactive plans.

References

Agre, P. and Chapman, D. (1987). Pengi: An implementation of a theory of activity. *Proceedings of the Sixth National Conference on Artificial Intelligence.*

Erickson, M. and Zytkow, J. (1988). Utilizing experience for improving the tactical manager. *Proceedings of the Fifth International Conference on Machine Learning.* Ann Arbor, MI.

Forbus, K. (1984). Qualitative process theory. *Artificial Intelligence*, 24(1-3). North-Holland Publishing Company, Amsterdam, The Netherlands.

Gervasio, M. and DeJong, G. (1989). Explanation-based learning of reactive operators. *Proceedings of the Sixth International Workshop on Machine Learning.* Ithica, NY. Morgan Kaufmann Publishers, Inc.

Grefenstette, J. (1988). Credit assignment in rule discovery system based on genetic algorithms. *Machine Learning*, 3(2/3). Kluwer Academic Publishers, Hingham, MA.

Grefenstette, J. (1989). A system for learning control strategies with genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms.* Fairfax, VA: Morgan Kaufmann.

Grefenstette, J., Ramsey, C. and Schultz, A. (1990). Learning sequential decision rules using simulation models and competition. To appear in *Machine Learning Journal.* Kluwer Academic Publishers, Hingham, MA.

Minton, S., Carbonell, J., Knoblock, C., Kuokka, D., Etzioni, O., and Gil, Y. (1989). Explanation-based learning: A problem-solving perspective. Carnegie-Mellon University Technical Report Number CMU-CS-89-103.

Mitchell, T. (1983). Learning by experimentation: Acquiring and refining problem-solving heuristics. In R. Michalski, J. Carbonell, and T. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach* (Vol. 1). Tioga Publishing Co., Palo Alto, CA.

Mitchell, T., Keller, R. and Kedar-Cabelli, S. (1986). Explanation-based generalization: A unifying view. *Machine Learning*, 1(1). Kluwer Academic Publishers, Hingham, MA.

Sammut, C. and Banerji, R. (1986). Learning concepts by asking questions. In R. Michalski, J. Carbonell, and T. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach* (Vol. 2). Morgan Kaufmann Publishers, Los Altos, CA.

Schoppers, M. (1987). Universal plans for reactive robots in unpredictable environments. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence.*

Schultz, A., Ramsey, C. and Grefenstette, J. (1990). Simulation-assisted learning by competition: Effects of noise differences between training model and target environment. In *Proceedings*

of the Seventh International Machine Learning Conference. Austin, TX: Morgan Kaufmann.